Standard Code Library

FLself

SCUT

June 1, 2022

Contents

一切的开始 宏定义	2
数据结构 ST 表	2
数学 类欧几里得	2
图论 LCA	3
计算几何 二维几何:点与向量	3
字符串 后缀自动机	4
杂项 STI.	4

一切的开始

宏定义

● 需要 C++11

```
#include <bits/stdc++.h>
    using namespace std;
   using LL = long long;
   #define FOR(i, x, y) for (decay < decltype(y) > :: type i = (x), _##i = (y); i < _##i; ++i)
   #define FORD(i, x, y) for (decay < decltype(x) > :: type i = (x), _##i = (y); i > _##i; --i)
   #ifdef zerol
    #define dbg(x...) do { cout << "\033[32;1m" << #x << " -> "; err(x); } while (0)
   void err() { cout << "\033[39;0m" << endl; }</pre>
    template<template<typename...> class T, typename t, typename... A>
   void err(T<t> a, A... x) { for (auto v: a) cout << v << ' '; err(x...); }</pre>
   template<typename T, typename... A>
11
    void err(T a, A... x) { cout << a << ' '; err(x...); }</pre>
   #else
13
   #define dbg(...)
   #endif
```

数据结构

ST 表

二维

```
int f[maxn][maxn][10][10];
    inline int highbit(int x) { return 31 - __builtin_clz(x); }
    inline int calc(int x, int y, int xx, int yy, int p, int q) {
        return max(
            \max(f[x][y][p][q], f[xx - (1 << p) + 1][yy - (1 << q) + 1][p][q]),
            \max(f[xx - (1 << p) + 1][y][p][q], f[x][yy - (1 << q) + 1][p][q])
7
   }
    void init() {
        FOR (x, 0, highbit(n) + 1)
        FOR (y, 0, highbit(m) + 1)
11
            FOR (i, 0, n - (1 << x) + 1)
12
            FOR (j, 0, m - (1 << y) + 1) {
13
                if (!x && !y) { f[i][j][x][y] = a[i][j]; continue; }
14
                f[i][j][x][y] = calc(
16
                    i, j,
                    i + (1 << x) - 1, j + (1 << y) - 1,
                    max(x - 1, 0), max(y - 1, 0)
18
19
                );
            }
20
21
   }
22
    inline int get_max(int x, int y, int xx, int yy) {
        return calc(x, y, xx, yy, highbit(xx - x + 1), highbit(yy - y + 1));
23
24
```

数学

类欧几里得

- $m = \lfloor \frac{an+b}{c} \rfloor$.
- $f(a,b,c,n) = \sum_{i=0}^{n} \lfloor \frac{ai+b}{c} \rfloor$: 当 $a \ge c$ or $b \ge c$ 时, $f(a,b,c,n) = (\frac{a}{c})n(n+1)/2 + (\frac{b}{c})(n+1) + f(a \bmod c, b \bmod c, c, n)$; 否则 f(a,b,c,n) = nm f(c,c-b-1,a,m-1)。
 $g(a,b,c,n) = \sum_{i=0}^{n} i \lfloor \frac{ai+b}{c} \rfloor$: 当 $a \ge c$ or $b \ge c$ 时, $g(a,b,c,n) = (\frac{a}{c})n(n+1)/(2n+1)/(6+(\frac{b}{c})n(n+1)/2 + g(a \bmod c, b \bmod c, c, n)$; 否则 $g(a,b,c,n) = \frac{1}{2}(n(n+1)m f(c,c-b-1,a,m-1) h(c,c-b-1,a,m-1))$ 。
- $h(a,b,c,n) = \sum_{i=0}^{n} \lfloor \frac{ai+b}{c} \rfloor^2$: $\stackrel{\triangle}{=} a \ge c \text{ or } b \ge c \text{ ff}, \ h(a,b,c,n) = (\frac{a}{c})^2 n(n+1)(2n+1)/6 + (\frac{b}{c})^2 (n+1) + (\frac{b}{c})^2 (n+1)$ $(\frac{a}{c})(\frac{b}{c})n(n+1)+h(a \bmod c,b \bmod c,c,n)+2(\frac{a}{c})g(a \bmod c,b \bmod c,c,n)+2(\frac{b}{c})f(a \bmod c,b \bmod c,c,n);$ 否则

图论

LCA

● 倍增

```
void dfs(int u, int fa) {
        pa[u][0] = fa; dep[u] = dep[fa] + 1;
2
        FOR (i, 1, SP) pa[u][i] = pa[pa[u][i - 1]][i - 1];
3
        for (int& v: G[u]) {
4
            if (v == fa) continue;
            dfs(v, u);
        }
   }
8
    int lca(int u, int v) {
10
11
        if (dep[u] < dep[v]) swap(u, v);</pre>
        int t = dep[u] - dep[v];
12
        FOR (i, 0, SP) if (t & (1 << i)) u = pa[u][i];
13
        FORD (i, SP - 1, -1) {
14
15
            int uu = pa[u][i], vv = pa[v][i];
            if (uu != vv) { u = uu; v = vv; }
16
17
        return u == v ? u : pa[u][0];
18
   }
19
```

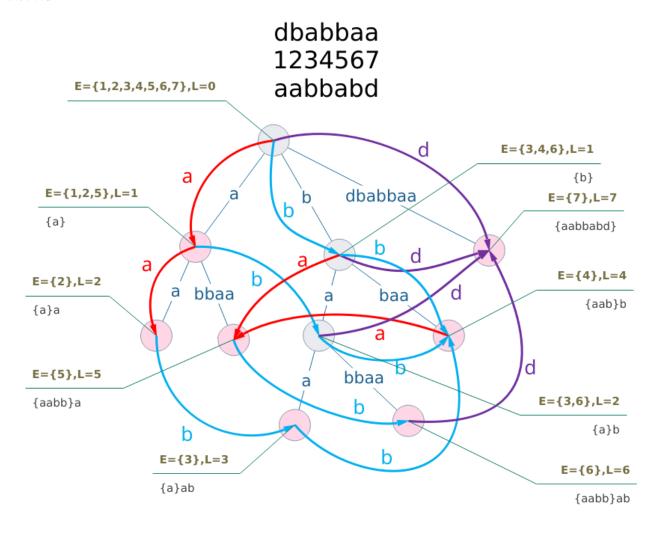
计算几何

二维几何: 点与向量

```
#define y1 yy1
    #define nxt(i) ((i + 1) % s.size())
    typedef double LD;
    const LD PI = 3.14159265358979323846;
    const LD eps = 1E-10;
    int sgn(LD x) { return fabs(x) < eps ? 0 : (x > 0 ? 1 : -1); }
    struct L;
    struct P;
    typedef P V;
    struct P {
10
11
        LD x, y;
        explicit P(LD x = 0, LD y = 0): x(x), y(y) {}
12
        explicit P(const L& l);
13
    };
14
    struct L {
15
16
        P s, t;
        L() {}
17
        L(P s, P t): s(s), t(t) {}
18
    };
19
20
    P operator + (const P& a, const P& b) { return P(a.x + b.x, a.y + b.y); }
21
    P operator - (const P& a, const P& b) { return P(a.x - b.x, a.y - b.y); }
22
    P operator * (const P& a, LD k) { return P(a.x * k, a.y * k); }
    P operator / (const P& a, LD k) { return P(a.x / k, a.y / k); }
    inline bool operator < (const P& a, const P& b) {</pre>
        return sgn(a.x - b.x) < 0 \mid \mid (sgn(a.x - b.x) == 0 \&\& sgn(a.y - b.y) < 0);
26
27
    bool operator == (const P& a, const P& b) { return !sgn(a.x - b.x) && !sgn(a.y - b.y); }
28
    P::P(const L& l) { *this = l.t - l.s; }
29
30
    ostream &operator << (ostream &os, const P &p) {
        return (os << "(" << p.x << "," << p.y << ")");
31
32
    istream &operator >> (istream &is, P &p) {
33
        return (is >> p.x >> p.y);
34
    }
35
36
```

字符串

后缀自动机



杂项

STL

copy

```
template <class InputIterator, class OutputIterator>
utputIterator copy (InputIterator first, InputIterator last, OutputIterator result);
```